

NSD Project1 DAY02

1. [案例1：网站架构演变](#)
2. [案例2：LNP+Mariadb数据库分离](#)
3. [案例3：Web服务器集群](#)

1 案例1：网站架构演变

1.1 问题

学习从单机架构到集群架构的演变之路：

- 单机版LNMP
- 独立数据库服务器
- Web服务器集群与Session保持
- 动静分离、数据库集群
- 各种缓存服务器
- 业务模型

1.2 步骤

此案例主要是学习网站架构演变的过程，以拓扑图和理论为主，具体实现还需要结合具体的软件。

步骤一：单机版LNMP

单机版网站，拓扑如图-1所示。

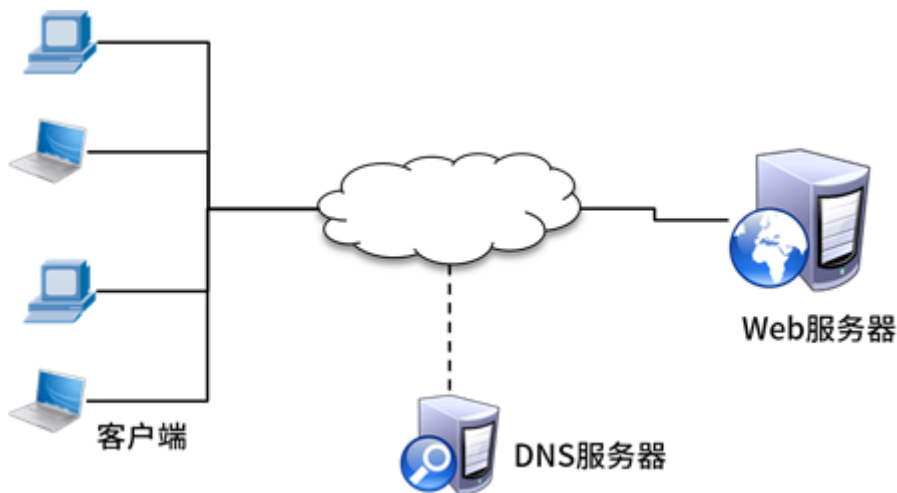


图-1 单机版网站服务器

用户量少时使用，简单、成本低、存在单点故障。

步骤二：独立数据库服务器

独立数据库服务器是将网站静态文件、代码文件等资料与数据库分离的架构，当用户量增加时单机的处理能力有限，PHP或JAVA代码的执行需要消耗大量CPU资源，数据库的增删改查需要调用大量的内存资源，将两者分离可以减轻服务器的压力，其拓扑结构如图-2所示。

[Top](#)

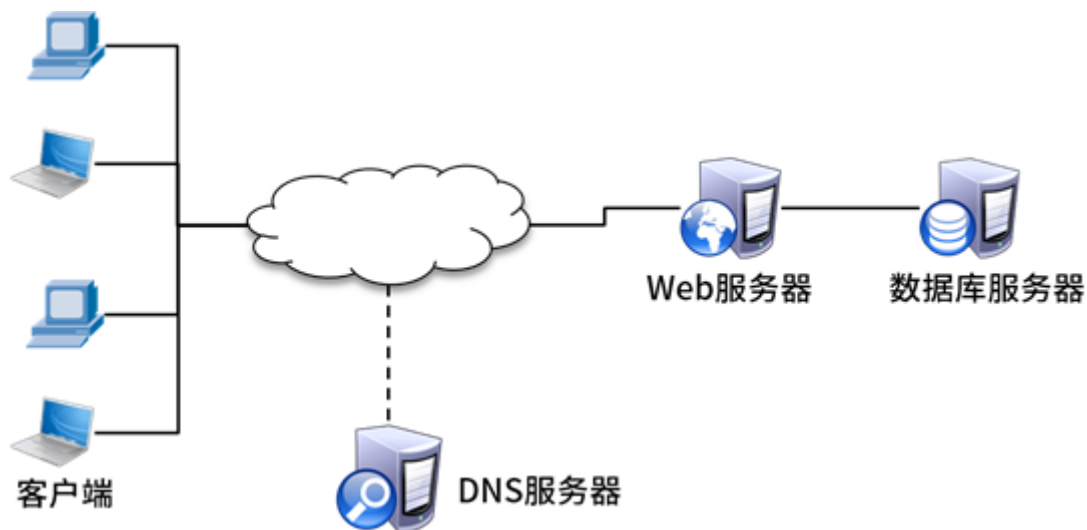


图-2 web服务器与数据库分离

Web服务器和数据库服务器的压力都可以得到有效改善，访问量有所增加。但是服务器依然存在单点故障问题。

步骤三：Web服务器集群与Session保持

我们可以通过Nginx、Haproxy代理服务器实现Web负载均衡集群，也可以使用LVS调度器实现Web负载均衡集群。部署完Web集群后还需要考虑如何进行Session会话保持，方法很多，如：根据源IP保持，代理服务器重写Cookie信息，共享文件系统保存session，使用数据库共享session等等。

该架构拓扑如图-3所示。

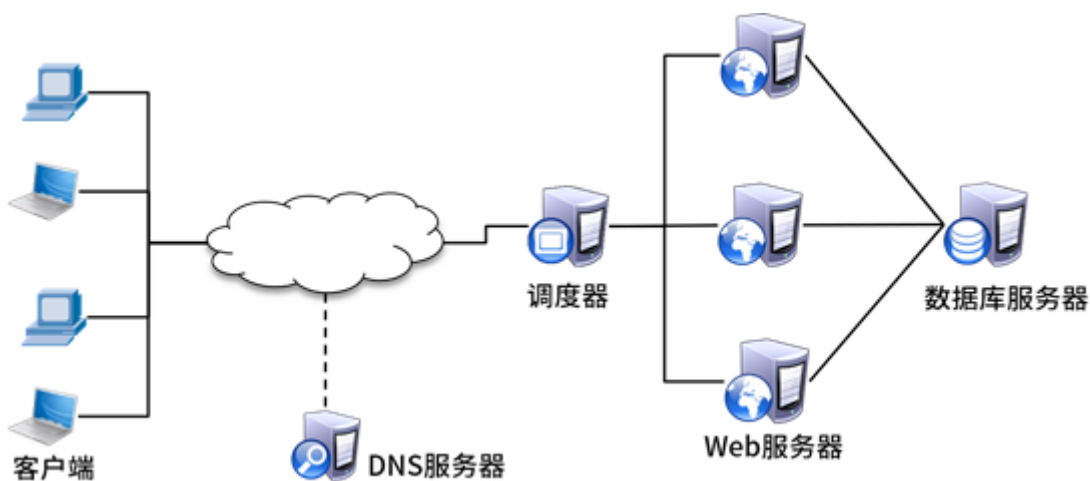


图-3

但是如果只有一台调度器依然会导致单点故障的问题，因此还需要使用Keepalived或Heartbeat之类的软件进行高可用配置，如图-4所示。

[Top](#)

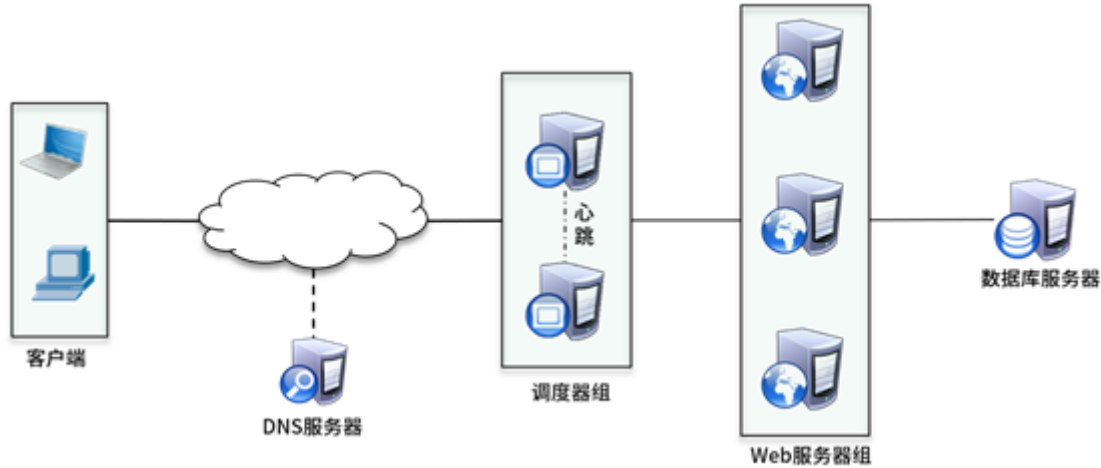


图-4

对于网站内容而言可以分离为动态页面和静态页面，静态页面就需要数据文件，动态页面则需要CPU解析代码，需要消耗大量的CPU资源，因此可以将静态和动态分离为两组服务器，动态页面有脚本代码组成，是一种基于网页的应用程序，因此这一组服务器也称为应用服务器，其架构如图-5所示。

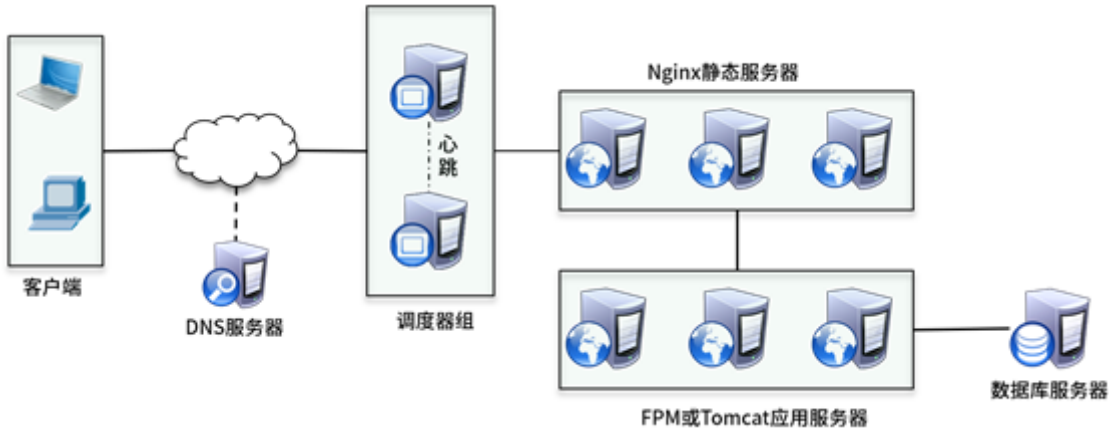


图-5

步骤四：动静分离、数据库集群

随着服务器的增加，虽然性能与并发量得到了明显的提升，但是数据的一致性、管理的便利性成为了新的问题，因此就需要增加统一的存储服务器，实现数据的同步一致，可以使用NFS，GlusterFS、Ceph等软件实现该功能，其架构如图-6所示。

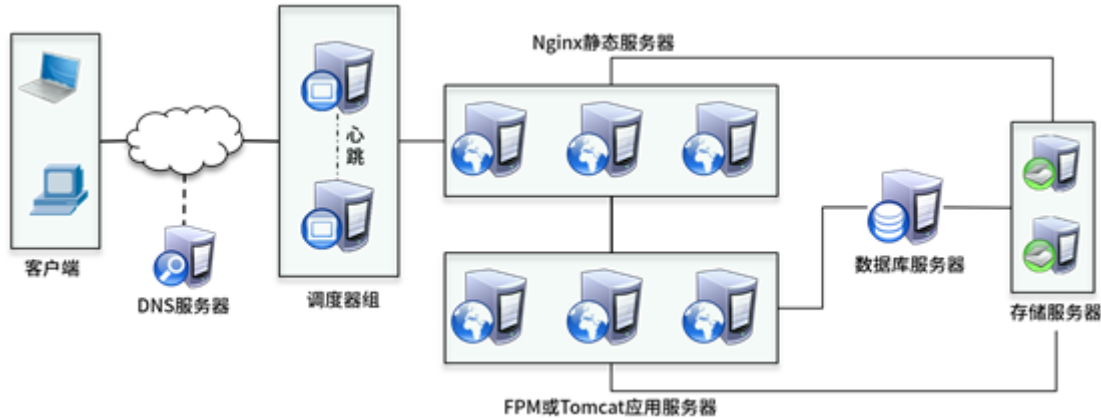


图-6

[Top](#)

此时所有应用服务器都连接一台数据库服务器进行读写操作，而且后期随着数据库中的数据不断增加，会导致数据库成为整个网站的瓶颈！这就需要对数据进行分库分表，创建数据库主从

或者数据库集群，实现读写分离，其拓扑如图-7所示。

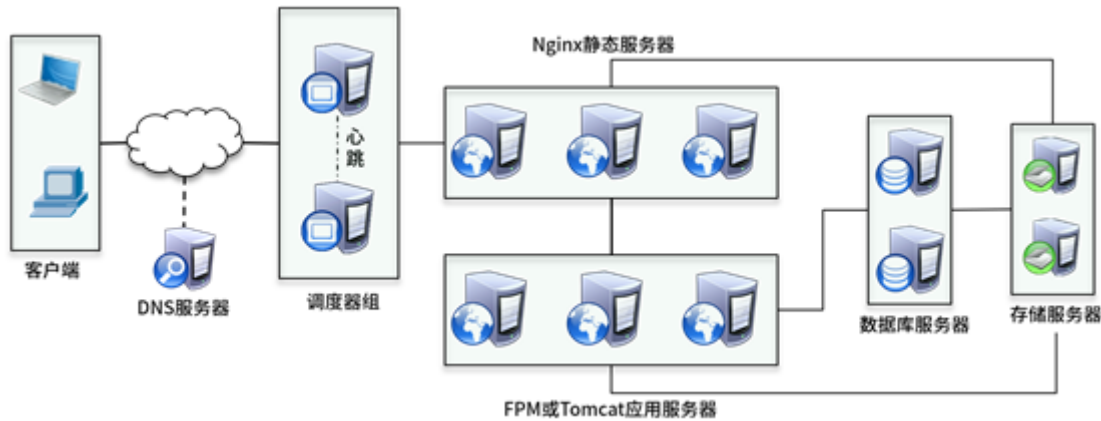


图-7

步骤五：缓存服务器与业务模型

对于静态数据我们可以通过varnish、squid或者nginx进行缓存，将数据缓存到距离用户更近的位置，构建CDN（内容分发网络）架构。

对于传统的SQL数据库而言，我们也可以通过增加NoSQL数据库，实现数据缓存的功能，提升数据库的访问速度。

备注：数据库相关知识在第三阶段课程有详细介绍，第二阶段项目暂时不做数据库优化。

最后，基于前面的架构，我们还可以将网站按照公司的业务进行分离，每个业务都可以是一个独立的集群，如图-8所示。



图-8

2 案例2：LNP+Mariadb数据库分离

2.1 问题

部署LNP+Mariadb实现数据库与Web服务器分离，实现以下目标：

- 将旧的数据库备份，迁移到新的服务器
- 修改配置调用新的数据库服务器

2.2 方案

实验拓扑如图-9所示，做具体实验前请先配置好环境。

[Top](#)



图-9

主机配置如表-1所示。

表-1

主机角色	IP 地址
client	private2 (192.168.2.254/24)
Web 服务器	eth1(192.168.2.11/24)
数据库服务器	eth1(192.168.2.21/24)

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署数据库服务器

1) 准备一台独立的服务器，安装数据库软件包

```

01. [root@database ~]# yum -y install mariadb mariadb-server mariadb-devel
02. [root@database ~]# systemctl start mariadb
03. [root@database ~]# systemctl enable mariadb
  
```

2) 将之前单机版LNMP网站中的数据库迁移到新的数据库服务器。

登陆192.168.2.11主机，备份数据库并拷贝给新的服务器，关闭旧的数据库服务。

```

01. [root@centos7 ~]# mysqldump wordpress > wordpress.bak
02. [root@centos7 ~]# scp wordpress.bak 192.168.2.21:/root/
03. [root@centos7 ~]# systemctl stop mariadb
04. [root@centos7 ~]# systemctl disable mariadb
  
```

登陆192.168.2.21主机，使用备份文件还原数据库。

创建空数据库：

```

01. [root@database ~]# mysql
02. MariaDB [(none)]> create database wordpress character set utf8mb4;
03. MariaDB [(none)]> exit
  
```

[Top](#)

使用备份文件还原数据：

```
01. [root@database ~]# mysql wordpress < wordpress.bak
```

重新创建账户并授权访问：

```
01. [root@database ~]# mysql
02. MariaDB [(none)]> grant all on wordpress.* to wordpress@'%' identified by 'v
03. MariaDB [(none)]> flush privileges;
04. MariaDB [(none)]> exit
```

3) 修改wordpress网站配置文件，调用新的数据库服务器。

Wordpress在第一次初始化操作时会自动生产配置文件：wp-config.php，登陆192.168.2.11修改该文件即可调用新的数据库服务。

```
01. [root@centos7 ~]# vim /usr/local/nginx/html/wp-config.php
02. 修改前内容如下：
03. define('DB_HOST', '192.168.2.11');
04. 修改后内容如下：
05. define('DB_HOST', '192.168.2.21');
```

步骤二：客户端测试

1) 客户端使用浏览器访问wordpress网站。

```
01. [root@client ~]# firefox http://192.168.2.11
```

3 案例3：Web服务器集群

3.1 问题

使用HAProxy部署Web服务器集群，实现以下目标：

- 部署三台Web服务器
- 迁移网站数据，使用NFS实现数据共享
- 部署HAProxy代理服务器实现负载均衡
- 部署DNS域名解析服务器

[Top](#)

3.2 方案

实验拓扑如图-10所示，做具体实验前请先配置好环境。

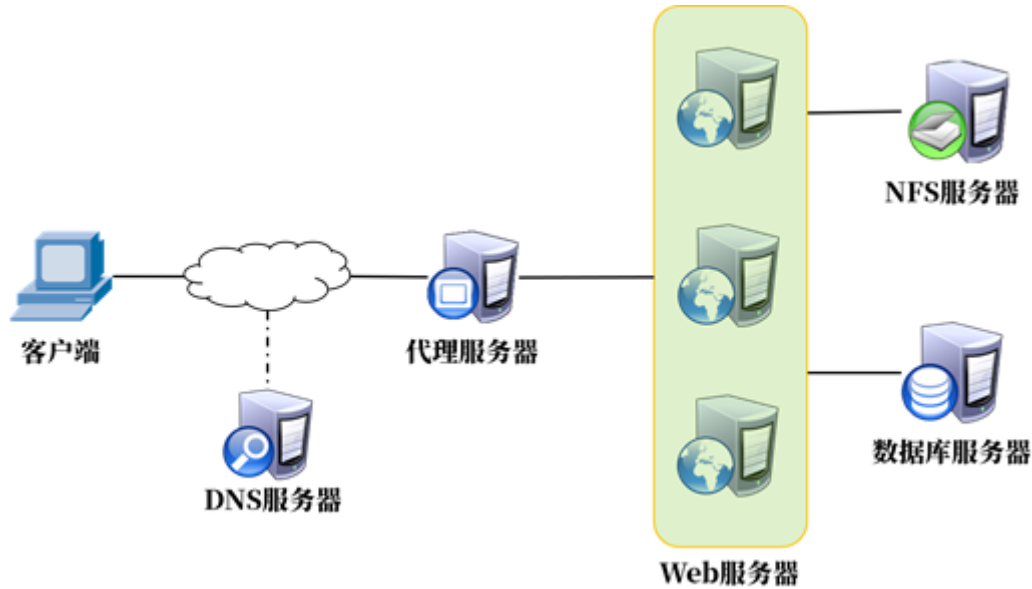


图-10

备注：实际操作中DNS服务代理服务器部署在同一台主机上（节约虚拟机资源）。
主机配置如表-2所示。

表-2

主机角色	主机名称	IP 地址
client	room9pc01	private2 (192.168.2.254/24)
代理服务器 DNS 服务器	proxy	eth0(192.168.4.5/24) eth1(192.168.2.5/24)
Web1 服务器	web1	eth1(192.168.2.11/24)
Web2 服务器	web2	eth1(192.168.2.12/24)
Web3 服务器	web3	eth1(192.168.2.13/24)
数据库服务器	database	eth1(192.168.2.21/24)
NFS 服务器	nfs	eth1(192.168.2.31/24)

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署web2和web3服务器

1) 安装LNP软件包

```
01. [root@web2 ~]# yum -y install gcc pcre-devel openssl-devel
02. [root@web2 lnmp_soft]# tar -xf nginx-1.12.2.tar.gz
03. [root@web2 lnmp_soft]# cd nginx-1.12.2/
04. [root@web2 nginx-1.12.2]# ./configure \
05. --with-http_ssl_module \
06. --with-http_stub_status_module
07. [root@web2 nginx-1.12.2]# make && make instal
08. [root@web2 ~]# yum -y install php php-fpm php-mysql mariadb-devel
```

[Top](#)

```
09.
10.
11. [root@web3 ~]# yum -y install gcc pcre-devel openssl-devel
12. [root@web3 lnmp_soft]# tar -xf nginx-1.12.2.tar.gz
13. [root@web3 lnmp_soft]# cd nginx-1.12.2/
14. [root@web3 nginx-1.12.2]# ./configure \
15. --with-http_ssl_module \
16. --with-http_stub_status_module
17. [root@web3 nginx-1.12.2]# make && make instal
18. [root@web3 ~]# yum -y install php php-fpm php-mysql mariadb-devel
```

2) 修改nginx配置实现动静分离 (web2和web3操作)

web2修改默认首页index.php，配置两个location实现动静分离。

```
01. [root@web2 ~]# vim /usr/local/nginx/conf/nginx.conf
02. location / {
03.     root    html;
04.     index  index.php index.html index.htm;
05. }
06.
07. location ~ /\.php$ {
08.     root            html;
09.     fastcgi_pass    127.0.0.1:9000;
10.     fastcgi_index   index.php;
11.     include         fastcgi.conf;
12. }
```

web3修改默认首页index.php，配置两个location实现动静分离。

```
01. [root@web3 ~]# vim /usr/local/nginx/conf/nginx.conf
02. location / {
03.     root    html;
04.     index  index.php index.html index.htm;
05. }
06.
07. location ~ /\.php$ {
08.     root            html;
09.     fastcgi_pass    127.0.0.1:9000;
10.     fastcgi_index   index.php;
```

[Top](#)


```

11.             include         fastcgi.conf;
12.             }

```

3) 启动相关服务

```

01. [root@web2 ~]# echo "/usr/local/nginx/sbin/nginx" >> /etc/rc.local
02. [root@web2 ~]# chmod +x /etc/rc.local
03. [root@web2 ~]# /usr/local/nginx/sbin/nginx
04. [root@web2 ~]# systemctl start php-fpm #启动php-fpm
05. [root@web2 ~]# systemctl enable php-fpm
06.
07. [root@web3 ~]# echo "/usr/local/nginx/sbin/nginx" >> /etc/rc.local
08. [root@web3 ~]# chmod +x /etc/rc.local
09. [root@web3 ~]# /usr/local/nginx/sbin/nginx
10. [root@web3 ~]# systemctl start php-fpm #启动php-fpm
11. [root@web3 ~]# systemctl enable php-fpm

```

附加知识：systemd!!!

源码安装的软件默认无法使用systemd管理，如果需要使用systemd管理源码安装的软件需要手动编写服务的service文件（编写是可以参考其他服务的模板文件）。以下是nginx服务最终编辑好的模板。

Service文件存储路径为/usr/lib/systemd/system/目录。

```

01. [root@centos7 ~]# vim /usr/lib/systemd/system/nginx.service
02. [Unit]
03. Description=The Nginx HTTP Server
04. #描述信息
05. After=network.target remote-fs.target nss-lookup.target
06. #指定启动nginx之前需要其他的其他服务，如network.target等
07.
08. [Service]
09. Type=forking
10. #Type为服务的类型，仅启动一个主进程的服务为simple，需要启动若干子进程的服务为
11. ExecStart=/usr/local/nginx/sbin/nginx
12. #设置执行systemctl start nginx后需要启动的具体命令.
13. ExecReload=/usr/local/nginx/sbin/nginx -s reload
14. #设置执行systemctl reload nginx后需要执行的具体命令.
15. ExecStop=/bin/kill -s QUIT ${MAINPID}
16. #设置执行systemctl stop nginx后需要执行的具体命令.

```

[Top](#)

```
17.  
18.  [Install]  
19.  WantedBy=multi-user.target
```

步骤二：部署NFS，将网站数据迁移至NFS共享服务器

1) 部署NFS共享服务器

```
01.  [root@nfs ~]# yum install nfs-utils  
02.  [root@nfs ~]# mkdir /web_share  
03.  [root@nfs ~]# vim /etc/exports  
04.  /web_share 192.168.2.0/24(rw,no_root_squash)  
05.  
06.  [root@nfs ~]# systemctl restart rpcbind  
07.  [root@nfs ~]# systemctl enable rpcbind
```

NFS使用的是随机端口，每次启动NFS都需要将自己的随机端口注册到rpcbind服务，这样客户端访问NFS时先到rpcbind查询端口信息，得到端口信息后再访问NFS服务。

```
01.  [root@nfs ~]# systemctl restart nfs  
02.  [root@nfs ~]# systemctl enable nfs
```

2) 迁移旧的网站数据到NFS共享服务器

将web1 (192.168.2.11) 上的wordpress代码拷贝到NFS共享。

```
01.  [root@web1 ~]# cd /usr/local/nginx/  
02.  [root@web1 nginx]# tar -czpf html.tar.gz html/  
03.  [root@web1 nginx]# scp html.tar.gz 192.168.2.31:/web_share/
```

登陆nfs服务器，将压缩包解压

```
01.  [root@nfs ~]# cd /web_share/  
02.  [root@nfs web_share]# tar -xf html.tar.gz
```

[Top](#)

3)所有web服务器访问挂载NFS共享数据。

```
01. [root@web1 ~]# rm -rf /usr/local/nginx/html/*
02. [root@web1 ~]# yum -y install nfs-utils
03. [root@web1 ~]# echo "192.168.2.31:/web_share/html /usr/local/nginx/html/ nfs
04. [root@web1 ~]# mount -a
05.
06. [root@web2 ~]# yum -y install nfs-utils
07. [root@web2 ~]# echo "192.168.2.31:/web_share/html /usr/local/nginx/html/ nfs
08. [root@web2 ~]# mount -a
09.
10. [root@web3 ~]# yum -y install nfs-utils
11. [root@web3 ~]# echo "192.168.2.31:/web_share/html /usr/local/nginx/html/ nfs
12. [root@web3 ~]# mount -a
```

步骤三：部署HAProxy代理服务器

1) 部署HAProxy

安装软件，手动修改配置文件，添加如下内容。

```
01. [root@proxy ~]# yum -y install haproxy
02. [root@proxy ~]# vim /etc/haproxy/haproxy.cfg
03. listen wordpress *:80
04.     balance roundrobin
05.     server web1 192.168.2.11:80 check inter 2000 rise 2 fall 3
06.     server web2 192.168.2.12:80 check inter 2000 rise 2 fall 3
07.     server web3 192.168.2.13:80 check inter 2000 rise 2 fall 3
08.
09. [root@proxy ~]# systemctl start haproxy
10. [root@proxy ~]# systemctl enable haproxy
```

步骤三：部署DNS域名服务器

1) 安装DNS相关软件（192.168.4.5操作）。

```
01. [root@proxy ~]# yum -y install bind bind-chroot
```

2) 修改主配置文件，添加zone。

[Top](#)

```

01. [root@proxy ~]# vim /etc/named.conf
02. options {
03.     listen-on port 53 { any; };           #服务监听的地址与端口
04.     directory      "/var/named";         #数据文件路径
05.     allow-query     { any; };           #允许任何主机访问DNS服务
06.     ... ..
07. };
08.
09. zone "lab.com" IN {                       #定义正向
10.     type master;
11.     file "lab.com.zone";
12. };
13.
14. #include "/etc/named.rfc1912.zones";     #注释掉改行
15. #include "/etc/named.root.key";         #注释掉改行
16.
17. [root@proxy ~]# named-checkconf /etc/named.conf #检查

```

3) 修改正向解析记录文件。

注意：保留文件权限。

```

01. [root@proxy named]# cp -p /var/named/named.localhost /var/named/lab.com.zone
02. [root@proxy named]# vim /var/named/lab.zone
03. $TTL 1D
04. @      IN SOA  @ rname.invalid. (
05.                                     0      ; serial
06.                                     1D     ; refresh
07.                                     1H     ; retry
08.                                     1W     ; expire
09.                                     3H )   ; minimum
10. @      NS     dns.lab.com.
11. dns    A      192.168.4.5
12. www   A      192.168.4.5

```

4) 启动服务

[Top](#)

```

01. [root@proxy named]# systemctl start named
02. [root@proxy named]# systemctl enable named

```

5) 客户端修改DNS解析文件

提示：做完实验修改回原始内容。

```
01. [root@room9pc01 data]# cat /etc/resolv.conf
02. # Generated by NetworkManager
03. search tedu.cn
04. nameserver 192.168.4.5
05. nameserver 172.40.1.10
06. nameserver 192.168.0.220
```

步骤四：修改wordpress配置文件

1) 修改wp-config.php

在define('DB_NAME', 'wordpress')这行前面添加如下两行内容：

```
01. [root@web3 html]# vim /usr/local/nginx/html/wp-config.php
02. define('WP_SITEURL', 'http://www.lab.com');
03. define('WP_HOME', 'http://www.lab.com');
```

如果不添加这两行配置，浏览器访问网站某个子页面后，URL会固定到某一台后端服务器不轮询。

附加知识（常见面试题）

1) 什么是灰度发布：

答：灰度发布（又名金丝雀发布）是指在黑与白之间，能够平滑过渡的一种发布方式。

让一部分用户继续用产品特性A，一部分用户开始用产品特性B，如果用户对B没有什么反对意见，那么逐步扩大范围，把所有用户都迁移到B上面来。灰度发布可以保证整体系统的稳定，在初始灰度的时候就可以发现、调整问题，以保证其影响度。灰度期：灰度发布开始到结束期间的这一段时间，称为灰度期。

2) DNS服务器有哪些种，其使用的端口为多少？

答：有 根DNS、一级DNS、二级DNS、三级DNS、缓存DNS

主DNS服务器、从DNS服务器

端口：53

3) 从日志/opt/bjca3/logs/ca_access.log中截取14点到16点的日志，将截取的日志导入到/tmp/ca_access.txt中，日志格式如下：

答：awk '\$4>"13:00:00"&&\$4<="16:59:00" ca_access.log

4) 监控检查，使用ping命令编写脚本来查询一组IP地址同时检测他们是否处于活跃状态。要求（range：192.168.1.200-192.168.1.220，一个IP发送4个ping包，ping的过程不能输出信息到终端）？

[Top](#)

答:

```
01.  #!/bin/bash
02.  for i in {200..220}
03.  do
04.  ping -c 4 -i 0.2 -W 1 192.168.1.$i &>/dev/null
05.  if [ $? -eq 0 ];then
06.      echo "192.168.1.$i is up"
07.  else
08.      echo "192.168.1.$i is down"
09.  fi
10.  done
```

5) 假设nginx的访问日志格式如下, 统计访问页面前10位的IP数?

答: `awk '{IP[$1]++} END{for(i in IP){print i,IP[i]}}' access.log |sort -n |tail -10`

6) 请列举出10个以上的你所知晓的SQL语句?

参考答案:

```
01.  insert  select  delete  update  create  show  drop  grant  revoke  load
```

7) 如何切换到某个数据库, 并在上面工作?

答: `use 库名;`

8) 列出数据库内的所有表?

答: `show tables;`

9) 如何删除表、删除数据库?

答: `drop table 表名;drop database 库名;`

10) 如何列出表' xrt' 内name域值为' tecmint' , web_address域值为' tecmint.com' 的所有数据?

答:

```
01.  select * from xrt where name=" tecmint" and web_address=" tecmint.com"
```

[Top](#)